

# Sisteme liniare - metode iterative

Radu T. Trîmbițaș

21 martie 2022

## 1 Metode iterative staționare

Dorim să calculăm soluția sistemului

$$Ax = b, \quad (1)$$

când  $A$  este inversabilă. Presupunem că am găsit o matrice  $T$  și un vector  $c$  astfel încât  $I - T$  este inversabilă și punctul fix unic al ecuației

$$x = Tx + c \quad (2)$$

coincide cu soluția sistemului  $Ax = b$ . Fie  $x^*$  soluția lui (1) sau, echivalent, a lui (2).

Iterația:  $x^{(0)}$  dat; se definește  $(x^{(k)})$  prin

$$x^{(k+1)} = Tx^{(k)} + c, \quad k \in \mathbb{N}. \quad (3)$$

Metoda (3) este convergentă dacă și numai dacă  $\rho(T) < 1$ , unde  $\rho(T)$  este raza spectrală a lui  $T$ .

Criteriul de oprire este

$$\|x^{(k)} - x^{(k-1)}\| \leq \frac{1 - \|T\|}{\|T\|} \varepsilon. \quad (4)$$

Presupunem că putem descompune  $A$  sub forma  $A = M - N$ . Dacă  $M$  este ușor de inversat (diagonală, triunghiulară, ș.a.m.d.) este mai ușor să realizăm calculele în modul următor

$$Ax = b \Leftrightarrow Mx = Nx + b \Leftrightarrow x = M^{-1}Nx + M^{-1}b$$

Ultima ecuație este de forma  $x = Tx + c$ , unde  $T = M^{-1}N = I - M^{-1}A$ . Se obține șirul

$$x^{(k+1)} = M^{-1}Nx^{(k)} + M^{-1}b, \quad k \in \mathbb{N}, \quad (5)$$

unde  $x^{(0)}$  este un vector arbitrar. Considerăm descompunerea  $A = D - L - U$ , unde

$$(D)_{ij} = a_{ij}\delta_{ij}, \quad (-L)_{ij} = \begin{cases} a_{ij}, & i > j \\ 0, & \text{altfel} \end{cases}$$

$$(-U)_{ij} = \begin{cases} a_{ij}, & i < j \\ 0, & \text{altfel} \end{cases}$$

Pentru diverse alegeri ale lui  $M$  și  $N$  se obține:

- metoda lui Jacobi:  $M = D$ ,  $N = L + U$ . În acest caz,  $T = D^{-1}(L + U)$ ,  $c = D^{-1}b$ . Scrisă pe componente, metoda are forma

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left( b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)} \right)$$

- metoda Gauss-Seidel:  $M = D - L$ ,  $N = U$ . În acest caz,  $T = (D - L)^{-1}U$ ,  $c = (D - L)^{-1}b$ . Scrisă pe componente, metoda are forma

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left( b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)} \right)$$

- metoda SOR (Successive OverRelaxation). În acest caz,  $M = \frac{D}{\omega} - L$ . Se obține

$$T = (D - \omega L)^{-1}((1 - \omega)D + \omega U),$$

$$c = \omega(D - \omega L)^{-1}b.$$

Pe componente,

$$x_i^{(k+1)} = (1 - \omega)x_i^{(k)} + \frac{\omega}{a_{ii}} \left( b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)} \right).$$

Valoarea optimă a lui  $\omega$ , valabilă doar pentru anumite tipuri de matrice (tridiagonale, tridiagonal pe blocuri, ordonate consistent, etc.) este

$$\omega_O = \frac{2}{1 + \sqrt{1 - \rho^2}}, \quad (6)$$

unde  $\rho$  este raza spectrală a matricei metodei lui Jacobi.

## 2 Considerații de implementare

Dacă putem descompune  $A$  sub forma  $A = M - N$ , unde  $M$  este inversabilă, iterația (5) se scrie sub forma

$$x^{(k+1)} = M^{-1} (Nx^{(k)} + b), \quad k \in \mathbb{N}.$$

Folosind operatorul `\` din MATLAB putem scrie iterația sub forma

$$\mathbf{x} = \mathbf{M} \setminus (\mathbf{N} * \mathbf{x} + \mathbf{b})$$

unde  $\mathbf{x}$  este aproximația precedentă, iar  $\mathbf{x}$  aproximația curentă.

Dacă  $\varepsilon_a$  este eroarea absolută, iar  $\varepsilon_r$  eroarea relativă, criteriul de oprire se poate scrie sub forma (criteriu mixt)

$$\|x^{(k)} - x^{(k-1)}\| \leq \frac{1 - \|T\|}{\|T\|} \varepsilon_a + \varepsilon_r.$$

Pentru efectivitatea criteriului, cei doi termeni  $\frac{1 - \|T\|}{\|T\|} \varepsilon_a$  și  $\varepsilon_r$  trebuie să aibă același ordin de mărime. La implementarea criteriului de oprire pute lua în considerare și reziduul

$$r^{(k)} = b - Ax^{(k)}$$

și norma sa  $\|r^{(k)}\|$ . Alegerea unui criteriu de oprire eneral este o problemă dificilă.

Metodele iterative dau rezultate bune dacă sunt aplicate sistemelor mari cu matrice rară sau structurată.

## 3 Probleme

**Problema 1** *Implementați metoda lui Jacobi în MATLAB.*

**Problema 2** *Implementați metoda SOR în MATLAB. Găsiți  $\omega$  optim utilizând (6). Atenție: aceasta nu este o metodă practică pentru calculul lui  $\omega_o$ ; ea are numai scop didactic.*

**Problema 3** *Rezolvați sistemele:*

$$\begin{bmatrix} 5 & -1 & 0 & 0 & \dots & & 0 \\ -1 & 5 & -1 & 0 & \dots & & 0 \\ 0 & -1 & 5 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & -\ddots & 0 & \vdots \\ \vdots & & \ddots & -1 & 5 & -1 & 0 \\ 0 & \dots & \dots & 0 & -1 & 5 & -1 \\ 0 & \dots & \dots & \dots & 0 & -1 & 5 \end{bmatrix} x = \begin{bmatrix} 4 \\ 3 \\ 3 \\ \vdots \\ 3 \\ 3 \\ 4 \end{bmatrix}$$

$$\begin{bmatrix} 5 & -1 & 0 & -1 & \dots & & \dots & 0 \\ -1 & 5 & -1 & 0 & -1 & & & \vdots \\ 0 & -1 & 5 & -1 & \ddots & \ddots & & \vdots \\ -1 & & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & -1 & \ddots & -1 & 5 & -1 & 0 & -1 \\ 0 & \dots & \ddots & 0 & -1 & 5 & -1 & 0 \\ 0 & \dots & & -1 & 0 & -1 & 5 & -1 \\ 0 & \dots & & & -1 & 0 & -1 & 5 \end{bmatrix} x = \begin{bmatrix} 3 \\ 2 \\ 2 \\ 1 \\ \vdots \\ 2 \\ 2 \\ 3 \end{bmatrix}.$$

cu toate metodele implementate.

**Problema 4** *Generați sisteme cu matrice diagonal dominante aleatoare ce au soluția  $[1, \dots, n]^T$  și rezolvați-le cu metodele Jacobi, Gauss-Seidel și SOR.*

## 4 Probleme suplimentare

**Problema 5** *Testați rutinele implementate pentru matrice rare de diverse dimensiuni și comparați timpii de execuție cu cei necesari pentru matrice dense.*

**Problema 6** *Pentru rutinele implementate generați curbe de convergență, adică curbe semilogaritmice care au pe abscisă numărul pasului curent, iar pe ordonată logaritmul normei reziduiului. (folosiți funcția MATLAB `semilogy`).*